**自定义镜像与仓库**

自定义镜像

docker commit

使用镜像启动器，在该容器基础上修改

另存为一个镜像

docker run -itd centos bash

docker ps

docker exec -it IDs bash修改（增删改数据、安装软件、修改配置文件等）

docker commit IDs name:label





docker images



dockerfile

dockerfile语法格式

FROM:基础镜像

MAINTAINER:镜像创建者信息

EXPOSE:开放的端口

ENV:设置变量

ADD:复制文件到镜像

RUN:制作镜像时执行的命令，可以有多个

WORKDIR：定义容器默认工作目录

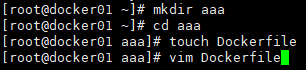
CMD:容器启动时执行的命令，仅可以有一条CMD，CMD [“/usr/bin/python”]

使用dockerfile工作流程

mkdir build;cd build

vim Dockerfile

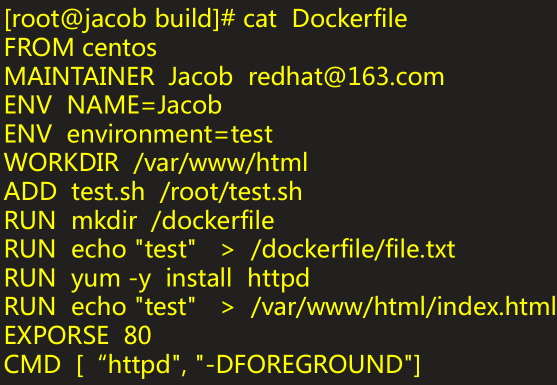
docker build -t imagename dockerfile所在的文件夹

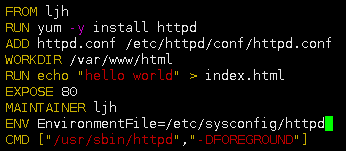






dockerfile文件案例

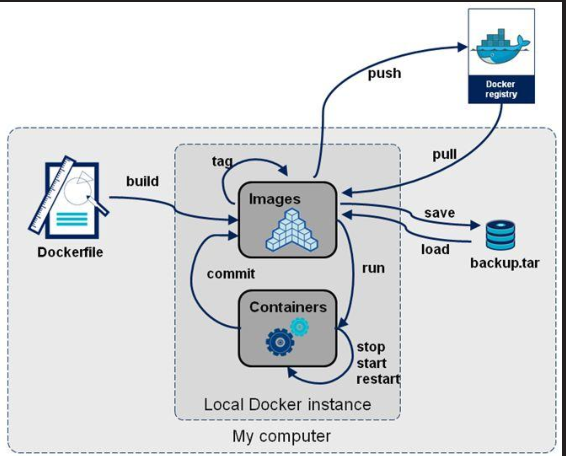




自定义镜像仓库

registry基本概念

共享镜像的一台服务器（镜像化的一台服务器）



自定义私有仓库

流程

docker pull registry

vim /usr/lib/systemd/system/docker.service

ExecStart=/usr/bin/dockerd --insecure-registry=ip:5000

systemctl daemon-reload

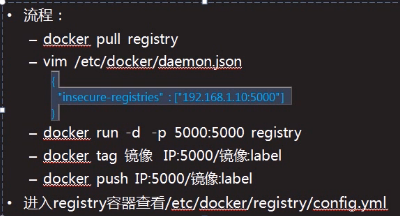
systemctl restart docker

docker run -id -p 5000:5000 registry

docker tag 镜像 IP:5000/镜像:label

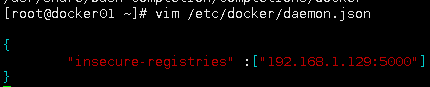
docker push IP:5000/镜像:label

进入registry容器查看/etc/docker/registry/config.yml



服务器配置私有仓库

编写配置文件



重启服务



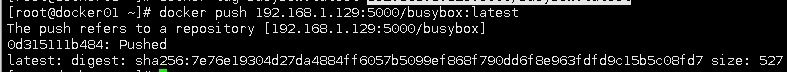
启动私有仓库服务



给镜像打标记

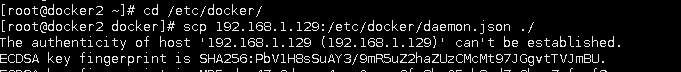


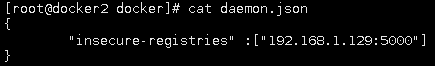
进行上传



客户端使用

编写配置文件

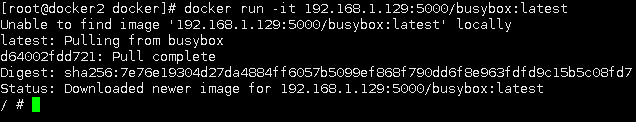




重启docker服务



启动容器，指定IP地址和端口，第一次要进行下载操作



查询私有仓库镜像

当前只有busybox一个镜像

显示镜像有几个标签



**持久化存储**

存储卷

卷的概念

docker容器不保持任何数据

重要数据请使用外部卷存储（数据持久化）

容器可以挂在真实机目录或共享存储为卷

主机卷的映射

将真实机目录挂在导容器中提供持久化存储，-v后头冒号前头是本机目录，冒号后是容器内目录，意思是用真机目录代替容器内目录

docker run -v /data:/data -it centos bash

共享存储

共享存储概念

一台共享存储服务器可以提供给所有docker主机使用

共享存储服务器（NAS、SAN、DAS等）

如：

使用NFS创建共享存储服务器

客户端挂载NFS共享，并最终映射到容器中

使用共享存储的案例

服务器

yum -y install nfs-utils

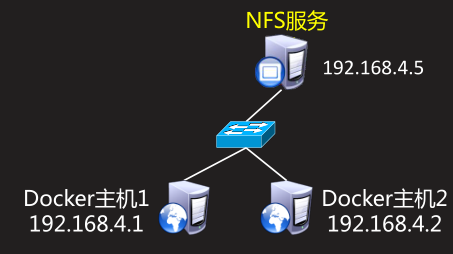
vim /etc/exports

systemctl start nfs

docker主机

mount挂载共享

运行容器时，使用-v选项映射磁盘到容器中



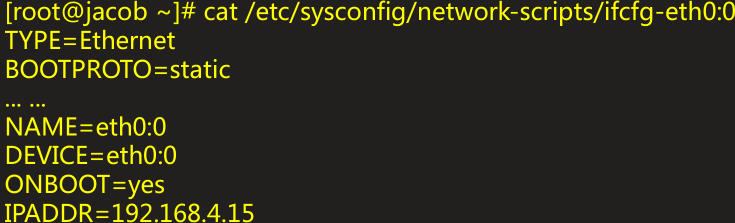
**docker网络架构**

linux网桥

创建虚拟网卡

真实网卡配置：cat /etc/sysconfig/networking-scripts/ifcfg-eth0

虚拟网卡配置：cat /etc/sysconfig/networking-scripts/ifcfg-eth0:0



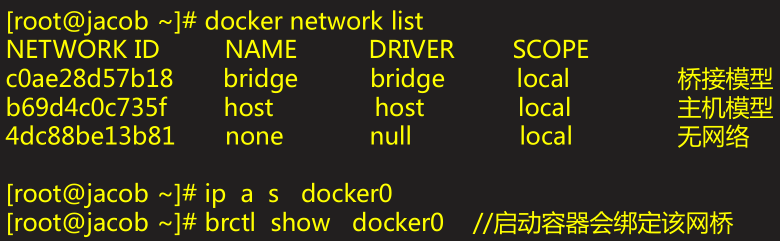
创建虚拟网桥



docker网络拓扑

查看docker默认网络模型

查看默认docker创建的网络模型



使用docker创建网桥

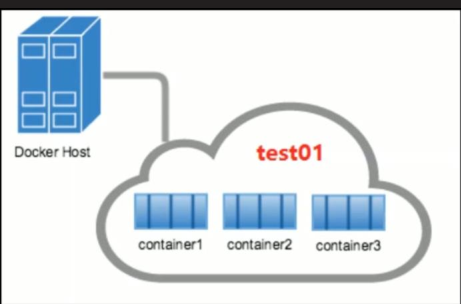
新建docker网络模型

docker network create --driver bridge test01

docker network list

ip a s

docker network inspect test01

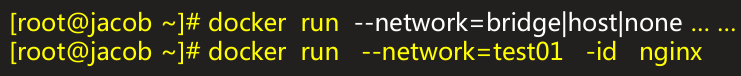


查看默认docker创建的网络模型（自定义网段）

docker network create --subnet=172.30.0.0/16 test01

使用自定义网桥

启动容器，使用刚刚创建的自定义网桥



客户端访问容器内的资源

默认容器通过SNAT可以访问外网

但外部网络的主机不可以访问容器内的资源

端口映射

使用端口映射可以实现外部网络访问容器内的资源

